A Rigorous Evaluation of Crossover and Mutation in Genetic Programming

(Or: how to lose your hair and sanity one experiment at a time)

David R White & Simon Poulding

University of York, UK

June 5, 2009

Outline

- Introduction to Genetic Programming

- Questioning Crossover

- Method

- Alternative Methodologies

- Results

What is GP?

GP field established in the early 90s. Depending on your point of view, it is a:

- ▶ Bio-inspired automated programming
- ▶ Machine-Learning technique
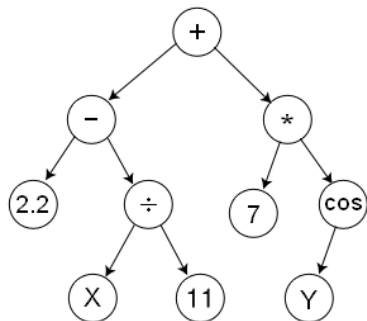- ▶ Form of heuristic search

What does GP do?

GP allows us to search a space of variable-length expressions.

Previous applications include:

- ▶ Designing analog circuits
- ▶ Performing symbolic regression
- ▶ Evolving sorting networks
- ▶ Discovery of quantum algorithms
- ▶ Optimising robot control
- ▶ Image processing
- ▶ Creating security protocols
- ▶ Data mining and classification
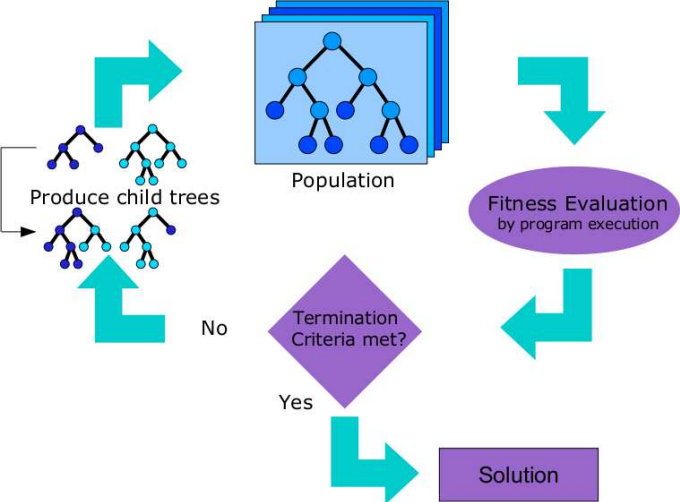- ▶ Game playing (Chess, Go)
- ▶ . . .

Tree-based GP

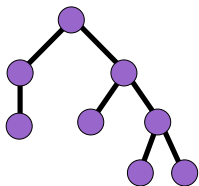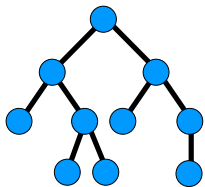Tree-based GP is the most popular amongst many alternatives (such as LinearGP, CartesianGP, Stack-Based GP).

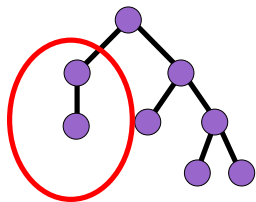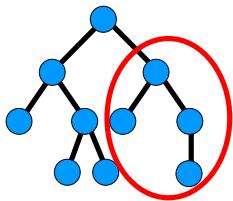$$\left(2.2 - \left(\frac{X}{11}\right)\right) + \left(7 * \cos(Y)\right)$$

GP: How does it work?

Population

Produce child trees
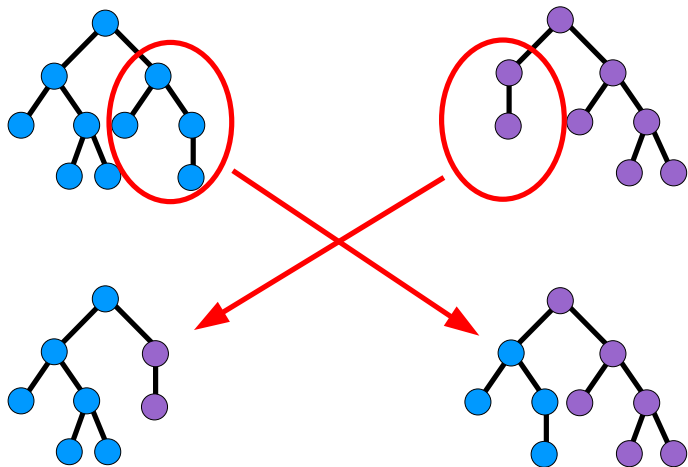
Fitness Evaluation
by program execution
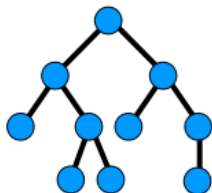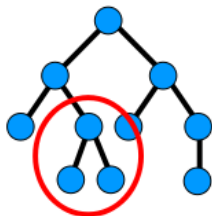
No

Termination Criteria met?
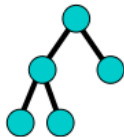
Yes

Solution

GP Crossover

GP Crossover

GP Crossover

GP Mutation

GP Mutation
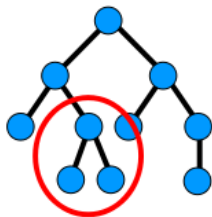
GP Mutation

GP Mutation

GP Mutation

Outline

Crossover in GP

What is the role of crossover in Genetic Programming?

- ▶ How does crossover work?
- ▶ Is it more useful than mutation?
- ▶ Should we be using crossover *at all*?

Theoretical Results

After 20 years of research we are still unable to answer these questions, despite some significant theoretical results [Langdon and Poli, 2002].

- ▶ Proposal: let's at least compare crossover and mutation empirically.

Theoretical Results

After 20 years of research we are still unable to answer these questions, despite some significant theoretical results [Langdon and Poli, 2002].

- ▶ Proposal: let's at least compare crossover and mutation empirically.
- ▶ Surely someone has done this before?

Empirical Investigation

Late 90s work on role of crossover:

- Angeline [1997a,b]
- Luke and Spector [1997, 1998]

Some systematic experimentation with different approaches.

Empirical Investigation

Late 90s work on role of crossover:

- ▶ Angeline [1997a,b]
- ▶ Luke and Spector [1997, 1998]

Some systematic experimentation with different approaches.

Results were inconclusive: revealed a complex picture.

Why are these questions so difficult?

Interaction Effects
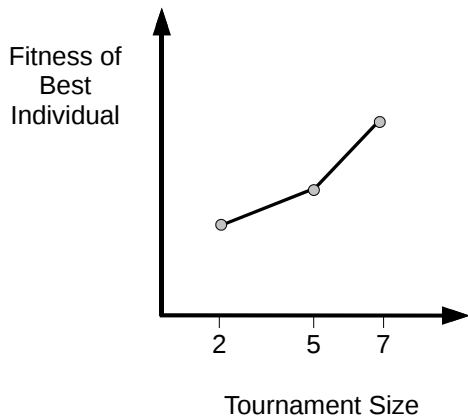
- Amongst parameter settings.
- Amongst problem characteristics.
- Between problem characteristics and parameter settings.
- Between random seed and the above.

Make it difficult to isolate the impact of a single factor on the success of the algorithm.
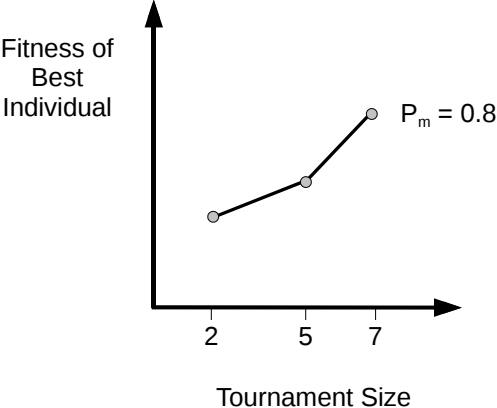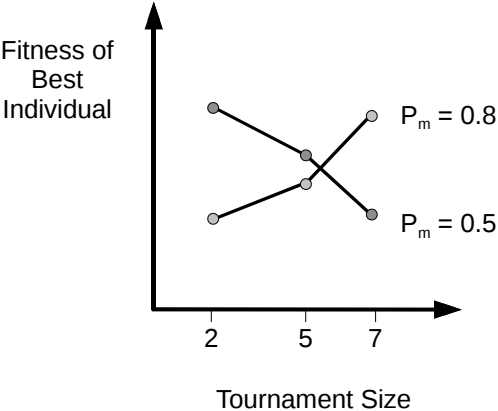
Interaction

Interaction

Interaction

Fitness of Best Individual

$P_m = 0.8$

2   5   7

Tournament Size
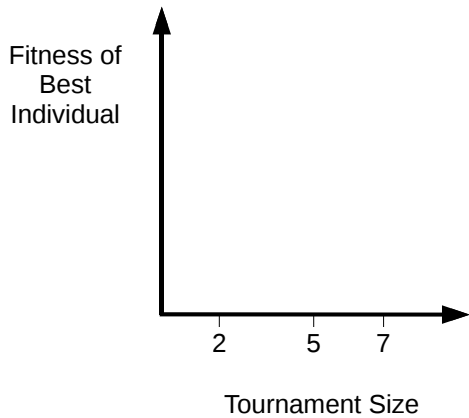
Interaction

Key Research Question

Which choice will yield the best results: the use of crossover *or* the use of mutation?

Key Research Question

Which choice will yield the best results: the use of crossover *or* the use of mutation?

How can we answer this question:

- ▶ In a fair, principled, rigorous manner.
- ▶ Taking into account the many parameters and stochastic nature of the algorithm.
- ▶ Such that we can state our results with a strong degree of confidence.

Outline

Challenge 1 – Ensure a Fair Comparison

How can we ensure a fair comparison?

GP using crossover

GP using mutation

Challenge 1 – Ensure a Fair Comparison

How can we ensure a fair comparison?

Victoria Sponge

Black Forest Gateau

Challenge 1 – Ensure a Fair Comparison

How can we ensure a fair comparison?

Victoria Sponge   Black Forest Gateau

Challenge 1 – Ensure a Fair Comparison

How can we ensure a fair comparison?

Victoria Sponge

Black Forest Gateau

Challenge 1 – Ensure a Fair Comparison

How can we ensure a fair comparison?

Victoria Sponge

Black Forest Gateau

Spend the same amount of time and effort perfecting each recipe

Challenge 2 – Determine The Perfect Recipe Objectively

How do I perfect the recipe for each cake?
I can't possibly try out all combinations of flour, eggs, cherries etc.

Try out a pre-determined, equivalent, finite set of recipes for each cake

Challenge 3 – Accommodate Random Variation

Each time I bake a cake using the same recipe, the result is slightly different.

Bake a number of cakes in order to 'average out' these differences
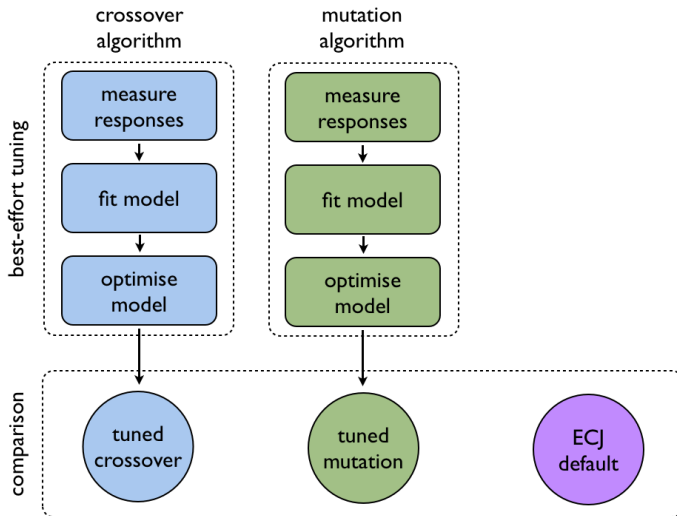
Challenge 4 – Make a Representative Comparison

I ask my grandma to try the cakes.
She doesn't like cherries.

Ask a more representative sample of people to try the cakes

Strategy

Problems

6 representative problems (distributed with ECJ):

- ▶ Symbolic Regression with no ERCs
- ▶ Symbolic Regression with ERCs
- ▶ Two Box Problem
- ▶ Santa Fe Ant Trail
- ▶ Boolean 11 Multiplexer
- ▶ Lawnmower

The entire method – tuning and comparison – is applied
independently to each problem.

Best-Effort Tuning 1 – Experimental Design

We considered 9 parameters for crossover, 10 for mutation.

Design points determined using 3-level factorial design: 3 levels chosen for each parameter, design points are combinations of these levels.

- crossover algorithm: $3^9 = 19,683$ design points
- mutation algorithm: $3^{10} = 59,049$ design points

Ran algorithm twice at each design point, and measured fitness of best individual in final population.

- ▶ Results are used to fit a second-order linear model that relates the response to the parameter values.
- ▶ The model is optimised to give tuned parameters.

Best-Effort Tuning 3 – Model Fitting and Optimisation
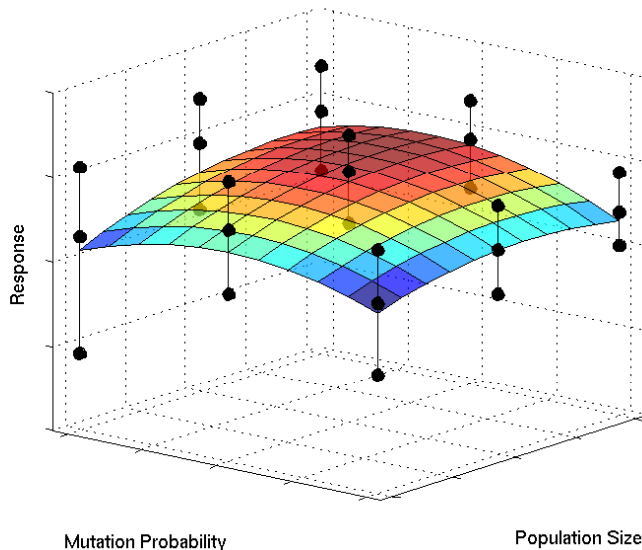
Algorithm Comparison

Ran each of the following algorithm 500 times:

- ▶ tuned crossover algorithm
- ▶ tuned mutation algorithm
- ▶ default ECJ (crossover) algorithm

Differences *between* algorithms are analysed for:

- ▶ statistical significance
  are differences unlikely to have occurred by chance?
- ▶ scientific significance
  are the differences large compared to the natural variance?

Statistical Significance

Applied rank-sum (Mann-Whitney-Wilcoxon) test:

- ▶ non-parametric - avoids the need to verify assumptions about the response distribution
- ▶ null hypothesis is that the responses from both algorithms are from the same distribution
- ▶ applied at the 5% significance level

Scientific Significance

Given a sufficiently large sample size, we would be able to demonstrate a statistically significant difference between the algorithm even if the difference were extremely small.

To guard against this possibility, we show that any difference in algorithm performance is also scientifically significant by analysing the effect size.

Vargha-Delaney Statistic

Calculated the Vargha-Delaney statistic:

- a non-parametric test
- statistic, A, takes values between 0 and 1
- easily calculated from rank-sum statistic
- $A = 0.5$ indicates no difference in algorithm performance
- values closer to 0 and 1 indicate that one algorithm or the other is better with increasingly large effect sizes
- we chose: scientific significance when $A < 0.36$ or $A > 0.64$
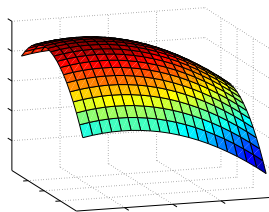- has a simple real-world interpretation

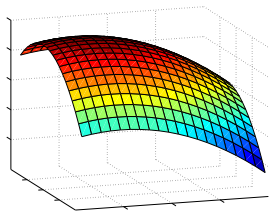# Outline

Alternative Methodologies

Our original intention was to use efficient experimental design methodologies:

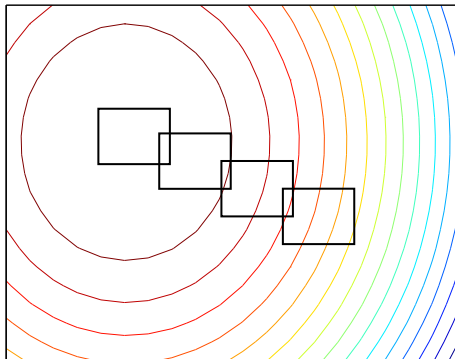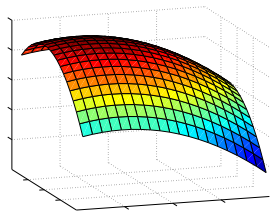- ▶ Response Surface Methodology
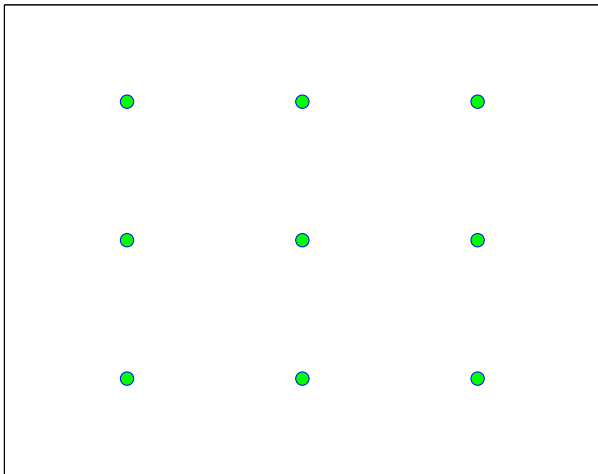- ▶ Central Composite Design

Response Surface Methodology
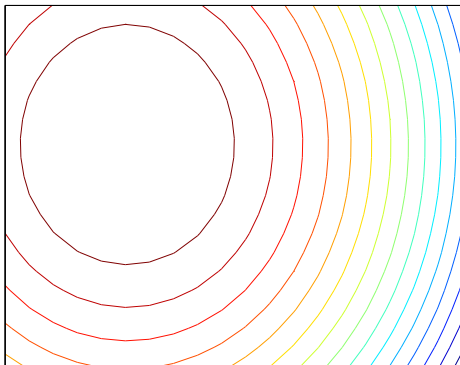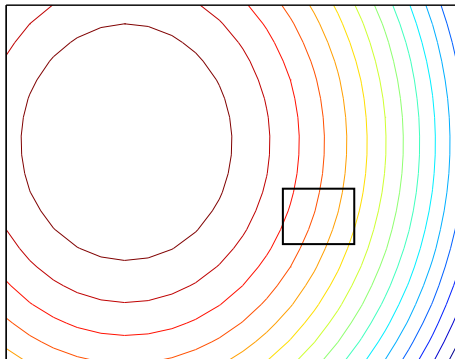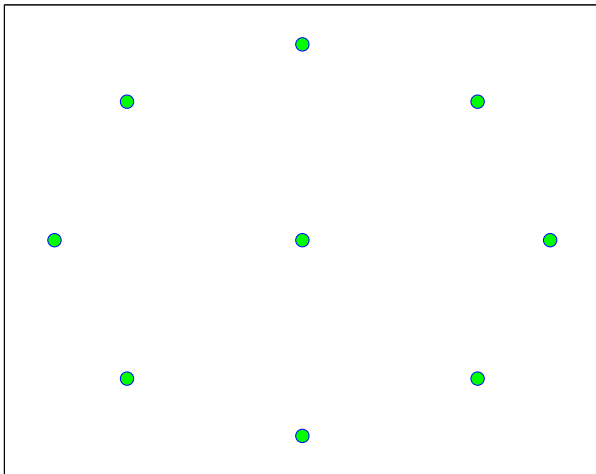
Response Surface Methodology

Response Surface Methodology

Central Composite Design

Central Composite Design

Central Composite Design

Full Factorial Design, 3 levels:
- crossover algorithm: $3^9 = 19,683$ design points
- mutation algorithm: $3^{10} = 59,049$ design points

Central Composite Design:
- crossover algorithm: 178 design points
- mutation algorithm: 178 design points

Outline

Results: Our question

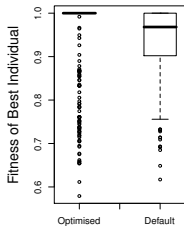Which choice will yield the best results: the use of crossover *or* the use of mutation?

Two steps:

1. Optimise the parameters of each algorithm on each problem.
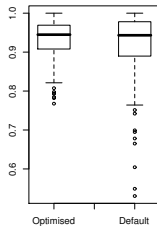2. Compare the two algorithms.

Parameter Tuning

- ▶ Our tuning method works.
- ▶ For half of the problems, we made a scientifically significant improvement to performance.
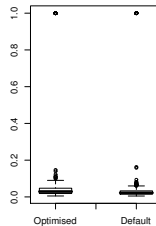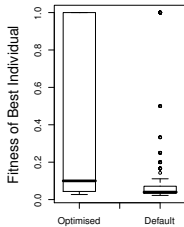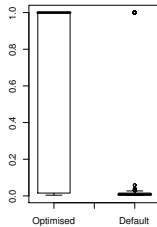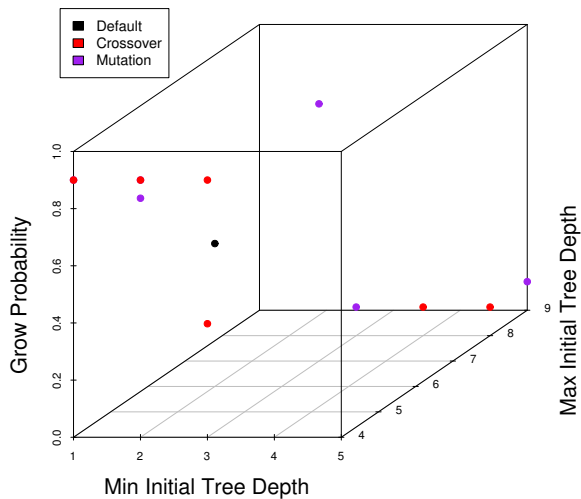
Tuning: Default versus Tuned
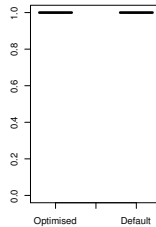
Example Optimal Parameters

Comparing Crossover to Mutation

- Crossover was significantly superior on only two out of six problems.
- These were a symbolic regression problem and the boolean 11 multiplexer.

. . . but crossover wasn't superior on the other symbolic regression problem.

Interpreting this Result

Crossover only works better than mutation one third of the time!

Interpreting this Result

Crossover is actually better than mutation one third of the time!

Interpreting this Result

Why is it better at all?

Luke and Spector

"The data is surprisingly complex. The difference between crossover and mutation is often small, and more often statistically insignificant. Further, where and why one is preferable to the other is strongly dependent on domain and parameter settings."

We can state that:

- ▶ In 2 cases the difference is scientifically significant.
- ▶ Our conclusions apply across parameter settings (within ranges).
- ▶ Problem characteristics remain as a strong factor.

Open Questions

Smaller questions:

- ▶ Values outside of our parameter ranges
- ▶ Other parameters, such as initialisation method

Larger questions:

- ▶ *Crossover works* – sometimes – but *why*?
- ▶ What will happen when we combine the two?

Open Questions: Experimental Method

Our approach used full factorial designs: robust, objective but inefficient.

- ▶ Fractional Factorial or Central Composite Designs would be more efficient.
- ▶ Response Surface Methodology could be considered when tuning the parameters.
- ▶ Which approaches are most suitable for GP parameters? Why?

. . . more hair to lose! (although approaching limiting case)

Repeating our Work

- ECJ is freely available
  http://cs.gmu.edu/~eclab/projects/ecj/
- Input data, results, source code available online
  http://www.cs.york.ac.uk/~drw/papers/eurogp2009/

Online Resources

Questions we received during the course of EuroGP

Do I have to run millions of experiments to use GP?

No! It is necessary here to provide a fair comparison when testing our hypotheses.

Why bother with this empirical malarky! Give me theory anyday!

The results here supply the theory with insight into which problems should be investigated.
We have concluded that any theoretical comparison between crossover and mutation *must* incorporate problem characteristics.

Questions we received during the course of EuroGP (Cont)

Haven't these case studies (Multiplexer etc.) been around forever? Why use these old problems?

These problems are everywhere in the literature, with no rigorous justification of why GP should solve them well. New problems would not have suited our aim of explaining why crossover works - people might simply claim that it doesn't on any proposed alternative case studies.

Have you found the optimal parameters for GP? Can we go now?

No! We only tuned for specific problems, and even then it as a "best effort" tuning process.

Your Questions?

Online

http://www.cs.york.ac.uk/~drw/papers/eurogp2009/

Acknowledgement

References I

Peter J. Angeline. Subtree crossover: Building block engine or macromutation? In Koza et al. [1997], pages 9–17.

Peter J. Angeline. Comparing subtree crossover with macromutation. In *EP '97: Proceedings of the 6th International Conference on Evolutionary Programming*, pages 101–112, London, UK, 1997b. Springer-Verlag. ISBN 3-540-62788-X. URL http://portal.acm.org/citation.cfm?id=738839.

John R. Koza, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max Garzon, Hitoshi Iba, and Rick L. Riolo, editors. *Genetic Programming 1997: Proceedings of the Second Annual Conference*, Stanford University, CA, USA, 1997. Morgan Kaufmann.

William B. Langdon and Riccardo Poli. *Foundations of Genetic Programming*. Springer-Verlag, 2002. ISBN 3-540-42451-2. URL http://www.springer.com/east/home?SGWID=5-102-22-2211943-0&changeHead

Sean Luke and Lee Spector. A comparison of crossover and mutation in genetic programming. In Koza et al. [1997], pages 240–248. URL http://cs.gmu.edu/~{}sean/papers/comparison/comparison.pdf.

Sean Luke and Lee Spector. A revised comparison of crossover and mutation in genetic programming. In John R. Koza, Wolfgang Banzhaf, Kumar Chellapilla, Kalyanmoy Deb, Marco Dorigo, David B. Fogel, Max H. Garzon, David E. Goldberg, Hitoshi Iba, and Rick Riolo, editors, *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pages 208–213, University of Wisconsin, Madison, Wisconsin, USA, 1998. Morgan Kaufmann. ISBN 1-55860-548-7. URL `http://cs.gmu.edu/~{}sean/papers/revisedgp98.pdf`.

Second Order Linear Model

$$y = \beta_0 + \sum_i \beta_i x_i + \sum_i \sum_{j>i} \beta_{ij} x_i x_j + \sum_i \beta_{ii} x_i^2 + \epsilon$$